Automatically Log Off Upon Disappearance of Facial Image

Xue Dong Yang
Department of Computer Science
University of Regina
Regina, Saskatchewan S4S 0A2

Peter Kort
Department of Computer Science
University of Regina
Regina, Saskatchewan S4S 0A2

Richard Dosselmann
Department of Computer Science
University of Regina
Regina, Saskatchewan S4S 0A2

Project Manager: S. Dahel 613-993-9949

Contract Number: W7714-4-0-9131

Contract Scientific Authority: Q. Xiao 613-998-1245

Defence R&D Canada - Ottawa

Contract Report
DRDC Ottawa CR 2005-051
March 2005

Abstract

The ability to prevent unauthorized users from gaining access to classified or sensitive data is crucial to many organizations, especially the military. There has been significantly increased attention in recent years in biometrics technology in order to provide more secured access controls to information systems. Different technologies and various products have been developed to authenticate users at entrances. However, once a user has logged on to a system, there is currently no mechanism to monitor the user's identity. It is important to ensure that the user, throughout the communication, is the authorized user for high-security applications. The objective of this project is to develop a demonstration system that will automatically log off a PC when the user's face disappears for an adjustable time interval.

In this report, a brief overview of face detection technologies is provided. The particular neural network-based face detection algorithm employed in the demo system is discussed in more detail. It is followed by the design of the demo system, and discussion of technical issues encountered during the development.

This page intentionally left blank.

Executive summary

In the last two decades, particularly after the Sept. 11 terrorist attacks, security has become the top priority on many governments' agenda around the world, including Canada. The ability to prevent unauthorized users from gaining access to classified or sensitive data is crucial to many organizations, especially the military. The need for heightened information security has expanded research focus from intrusion detection and virus protection by preventing unauthorized hackers, to much more sophisticated access control by authenticating individual users.

Several technologies and a number of products have been developed to authenticate users and to provide access control to data objects. Among them, biometrics, a promising technology for personal identification, has received ever-increasing attention as a potential solution to a wide range of problems. However, once a user has logged on to a system, there is currently no mechanism to monitor the user's identity. This security gap must be filled. It is important to ensure that the user, throughout the communication, is the authorized user for high-security applications. The objective of this project is to develop a demonstration system that will automatically log off a PC when the user's face disappears for an adjustable time interval.

Among the fundamental technologies of biometrics, facial recognition is the only one that can perform both functions — user authentication and user monitoring. Face recognition is a non-intrusive technique and people generally do not have any problem accepting it as an authentication/monitoring tool. Although there are various commercial products on face recognition, none satisfies our special requirements. These products are generally designed for user authentication or identification, but not for user monitoring. The proposed demonstration tool has been built using a commonly used webcam and specially designed real-time PC software to perform face detection and face tracking continuously after a user logs on, and log off the PC automatically when the user's face disappears for an adjustable time interval. It is simple, fast, and inexpensive, which are the major differences between the proposed project and the current facial recognition products.

In this report, a brief overview of face detection technologies is provided. The particular neural network-based face detection algorithm employed in the demo system is discussed in more detail. It is followed by the design of the demo system, and discussion of technical issues encountered during the development. Due to both the limited three-month time frame and the budget for this project, the demo system is restricted to face detection and real-time monitoring. The completed demo system has met all the pre-specified requirements and even exceeded some of them. A logical next step is to include face recognition into the system. The enabling technology developed by this project is not limited to the particular application of information system security related to networked PC users; it is applicable in many other situations in which both operator authentication and dynamic monitoring are required.

Yang, X., Kort, P. and Dosselmann, R. 2005. Automatically Log Off Upon Disappearance of Facial Image. DRDC Ottawa CR 2005-051 University of Regina.

Table of contents

List of figures

List of tables

Acknowledgements

This project was conducted at the New Media Studio Lab (NMSL), at the University of Regina, and was funded by a CFI grant.

This page intentionally left blank.

1. Introduction

In the last two decades, particularly after the Sept. 11 terrorist attacks, security has become the top priority on many governments' agenda around the world, including Canada. Among all of the measures undertaken, biometrics, a promising technology for personal identification, has received ever-increasing attention as a potential solution to a wide range of problems including everything from preventing further terrorist attacks to making our streets safer from domestic crime. The specific issue addressed by this project is the information system security related to networked PC users. However, the technology developed by this project will not be limited to this particular application, and would be potentially applicable to many others.

The need for heightened information security has expanded research focus from intrusion detection and virus protection by preventing unauthorized hackers, to much more sophisticated access control by authenticating individual users. The ability to prevent unauthorized users from gaining access to classified or sensitive data is of great importance to any organization, especially the military. Several technologies and a number of products have been developed to authenticate users and to provide access control to data objects. But once the user has logged on to a system, there is currently no mechanism to monitor the user's identity. This security gap must be filled. To provide a secure IT operating environment, the entity that is really at the other end of the network link not only must be identified, but also has to be monitored throughout the information transmission.

1.1 Objective of the project

The objective of this project, as described in the "Statement of Work", is descried below:

> *The objective of this contract is to develop a demonstration system that will automatically log off PC's when the user's face disappears for an adjustable time interval. The resulting demonstration can be further developed with facial matching algorithm as a user monitoring system. The final system can be used in high-security environment to ensure that the user logging onto the system is the same person.*

Among the fundamental technologies of biometrics, facial recognition is the only one that can perform both functions — user authentication and user monitoring. A typical face recognition system captures images with a digital camera and processes the images using a PC. Face recognition is a non-intrusive technique and people generally do not have any problem accepting it as an authentication/monitoring tool. Although there are various commercial products on face recognition, none satisfies our special requirements. These products are generally designed for user authentication or identification, but not for user monitoring. The proposed demonstration tool will be built using a commonly used webcam and specially designed real-time PC software to perform face detection and face tracking continuously after a user logs on, and log off the PC automatically when the user's face disappears for an adjustable time interval. It must be simple, fast, and inexpensive, which are the major differences between the proposed project and the current facial recognition products.

1.2 Detailed statement of work

The specific work defined in the "Statement of Work" includes, but is not limited to, the following items:

- Review the current stages of face detection and face tracking technologies and present a brief summary of major existing techniques.
- Develop an algorithm to capture the video stream from a Logitech QuickCam Orbit.
- Develop a frame grabber to obtain still images from the video stream.
- Develop an algorithm to locate the face in the captured image.
- Develop a GUI that can perform the image capture and face locate tasks, accept variables, and display the original and processed images.
- Deliver a demonstration system that shall lock the PC or log off the user when a face cannot be detected for an adjustable time interval.

Though a lot of preliminary research was started before January 2005 by the authors, the design and development of this demo system officially started on January 4, 2005, when the projected was awarded. A kick-off meeting was held on January 13, 2005, at the New Media Studio Lab of the University of Regina. The meeting was attended by the representative from DRDC, Dr. Qinghan Xiao, and the three authors of this report. The progress of the project was reported to DRDC on a bi-weekly basis. The developed demo system was demonstrated at and delivered to DRDC on March 24, 2005. This system meets all of the functions specified above, plus several additional features, such as multiple face detection, face distance estimation, etc.

The rest of the report is organized as the follows. Section 2 provides a brief overview of state-of-the-art techniques for face detection. The specific algorithm employed in this demo system is described in more detail in Section 3 along with the design of the prototype demo system. The technical issues encountered during implementation are discussed in Section 4. The summary of the project is given in the last section.

2. Overview of Face Detection Methods

After nearly 30 years of research, face recognition, an important application of image analysis and understanding, has started to find practical commercial and law enforcement applications. Most existing applications are related to personal identification at certain specific checkpoints, such as a physical entrance to a secured facility, or an electronic entrance to a secured information system. In these applications, other biometric personal identification technologies, such as fingerprint analysis, retinal or iris scans, can be, and often are used. However, the tools based on these technologies require a participant's explicit cooperation or knowledge, thus imposing restrictions on the applicability of the technologies. Furthermore, fingerprint analysis and retinal or iris scans are not suitable for the purpose of continuous monitoring (the specific interest of this project), because it is impractical to ask an operator to keep his/her finger(s) on a sensor during a process, or to have his/her eye(s) wide open and facing the scanner all of the time. On the other hand, a personal identification system based on analysis of frontal or profile images of a face is generally effective without the participant's cooperation or knowledge, and is much better suited for the purpose of continuous monitoring. The existing facial recognition technologies, however, have various limitations; thus, they are not yet sufficiently robust and reliable for all related applications. But, the challenges that have arisen from this area continue to attract intensive research efforts from several related disciplines such as image processing, pattern recognition, neural networks, computer vision, computer graphics and psychology. An excellent overview and critical analysis of state-of-the-art face recognition techniques can be found in [1].

Automatic face detection from still images or video frames is a crucial first step in any application that involves facial image analysis. For example, in a face recognition application, before a face can be compared to faces stored in a database, it must be extracted from a visual scene that may be very complex. Robust real-time detection of faces is needed in a video surveillance system where dynamic scenes are scanned for known faces. Though the problem of face recognition is considered to be difficult, the problem of face detection is certainly not trivial either. Real-time face detection is a very challenging problem itself, mainly due to the following difficult issues:

(1) A face can appear in an image at an arbitrary location, with an arbitrary orientation, and at an arbitrary scale.

(2) The background of an image can be highly complex.

(3) The appearance of faces is directly affected by a person's facial expression.

(4) In the case of color images, skin color varies from person to person, and varies under different illumination conditions, and thus cannot simply be used for face region extraction.

(5) Certain applications have gray-scale images only.

Clearly, a robust face detection technique must search the entire image and consider all possible scales, orientations, and illumination conditions. Often times, all of the computation has to be done in real-time.

Numerous efforts have been made in fast detection of faces from a complex scene, and many techniques have been developed. According to a recent comprehensive survey ([2]), face detection methods can be broadly classified into four categories: knowledge-based, feature-based, template matching, and appearance-based methods. In the following, a brief discussion is given for each category, and selected examples from each category are reviewed in more detail, along with their performance evaluation, if they are available in the literature. The purpose of this chapter is to provide comprehensive background knowledge in the field of face detection to the readers.

2.1 Knowledge-based methods

Human knowledge of what constitutes a typical face is translated into rules, usually in form of relationships between facial features. This type of technique typically employs a top-down strategy. It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a frontal face usually appears in an image with two symmetric eyes, a nose, and a mouth. The relationships between features can be represented by their relative distances and positions. However, it may be difficult to translate human knowledge into well-defined rules. If the rules are detailed and strict, they may fail to detect faces that do not adhere to all of the rules. If the rules are too general, they may give many false positives. Some example techniques from this category are briefly described below.

A hierarchical knowledge-based method was proposed by Yang and Huang [3]. Their system consists of three levels of rules. A multi-resolution hierarchy of images is created by averaging and sub-sampling. Examples of the coded rules used to locate face candidates in the lowest resolution include: “the center part of the face has four cells with a basically uniform intensity,” “the upper round part of a face has a basically uniform intensity,” and “the difference between the average gray values of the center part and the upper round part is significant.” At the next level, local histogram equalization is performed on the face candidate regions, followed by edge detection. Another set of rules at the last level examines facial features such as eyes and mouth. Evaluated on 60 images, this system located faces in 50 test images with 28 images in which false alarms appear. Although its detection rate is not very high, one attractive feature of this method is the coarse-to-fine strategy that reduces the required computation.

Kotropoulos and Pitas [4] presented another rule-based localization method. Given an intensity image, $I(x,y)$, of m columns and n rows, the horizontal and vertical profiles are obtained by a projection method respectively. The two local minima in the horizontal profile are said to correspond to the left and right side of the head, and the local minima in the vertical profile are determined for the location of mouth lips, nose tip, and eyes. This method achieved an 86.5 percent detection rate with test images containing a single face in a uniform background. It, however, performs poorly when attempting to locate a face in a complex background or when multiple faces are present.

2.2 Feature-based methods

In contrast to the top-down approach used by knowledge-based methods, the bottom-up strategy is usually employed in feature-based approaches. Typical facial features include eyebrows, eyes, nose, mouth, and hair-line, all of which can be extracted using edge detection methods. A statistical model can be built-up using the extracted features to describe their relationships for identification of faces. Structural features that exist even when the pose, viewpoint, or lighting conditions vary form the basis for face localization. One general problem with feature-based methods is that extracted boundaries are strongly affected by illumination conditions (such as shadows) and the edge detection results are often noisy.

There are many methods in this category. Two examples are given below that illustrate the principles involved. Heuristics (Sirohey [5]) can be used to remove and group edges so that only the ones on the face contour are preserved. An ellipse is then fit to the boundary between the head region and the background. This algorithm showed 80 percent accuracy with 48 testing images containing complex backgrounds. Band pass filtering and morphological operations (Graf et al [6]) can also be applied to enhance regions of high intensity that have certain shapes (e.g. eyes). Based on the peak value in the histogram and its width, adaptive threshold values can be selected to generate two binary images. Connected components in the binary images are identified for candidate facial features. Finally, classifiers are used to verify whether the combinations of such areas form faces.

Skin color is also a favorite feature in various other literatures. Although skin color varies from person to person, several studies suggested that the major difference is in their intensity rather than their chrominance (e.g. [6]). A very simple method (Chai and Ngan [7]) defines a region of skin tone pixels using *Cr* and *Cb* (in YCrCb color space). With carefully chosen intervals [*Cr1, Cr2*] and [*Cb1, Cb2*], a pixel belongs to skin tone if its values (*Cr, Cb*) fall with the range. Gaussian density functions and mixture of Gaussians are also often used to model distributions of skin colors. It should be noted that the above skin color models are not effective when the spectrum of light source varies significantly. Therefore, skin color alone is usually not sufficient to detect or track faces.

2.3 Template matching methods

The correlations between an input image and the stored standard patterns are computed. The existence of a face is determined based on correlation values. This approach is simple to implement, but has proven to be inadequate when dealing with variations in scale, pose, and shape. Therefore, multiresolution, multiscale, subtemplates, and deformable templates have been proposed to address these issues. In the following, we examine a few selected example techniques to have a basic understanding to this type of methods.

An early attempt (Sakai et al [8]) used several subtemplates for the eyes, nose, mouth and face contour to detect a frontal face. Each subtemplate is defined in terms of line segments. Lines extracted from the input image are matched against the subtemplates. The candidate locations of faces are first selected, and in the second phase, other subtempaltes are used to examine the details.

A two-stage face detection method proposed by Govindaraju et al [9] involves face hypotheses generation and testing. A face model consists of curves of the left side, the hair-line, and the right side of a frontal face. A cost is assigned to edges that form feature curves. If the cost of a group of three feature curves is low, the group becomes a hypothesis. Their system reports a detection rate of approximately 70 percent based on a set of 50 test photographs, all frontal upright faces.

A quantitative model for face pattern (QMF) is proposed by Tsukamoto et al [10]. In this model, each sample image is divided into a number of blocks, and quantitative features are estimated for each block. A face pattern is parameterized to measure "faceness" at every position of an input image.

A set of basis face silhouettes, the so called eigen-silhouettes obtained using principal component analysis (PCA) on face samples, have also been used as templates for face localization ([11]). The silhouette is represented by an array of bits. A generalized Hough transform is applied to the silhouette for face localization.

More sophisticated methods involve a hierarchical template (e.g. Miao et al [12]). A multiresolution image hierarchy is formed and edges are extracted. The face template consists of the edges produced by facial components, such as two eyebrows, two eyes, nose, and mouth. Finally, heuristics are applied to determine the existence of a face.

Parameterized deformable templates (e.g. Yuille et al [13]) can fit an a priori elastic model to facial features. An energy function is defined to link edges, peaks, and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters. Their experimental results demonstrate good performance in tracking non-rigid features.

2.4 Appearance-based methods

In contrast to template matching methods where templates are predefined by experts, the models in appearance-based methods are learned from a set of training images that are supposed to capture representative facial appearance. The learned characteristics are in the form of distribution models or discrimination functions. Therefore, many of them can be understood in a probabilistic framework.

Eigenface is, perhaps, the most well-known model for both face detection and face recognition. Article [14] (Kohonen) is an early example of this method. An autocorrelation matrix is formed for aligned and normalized face images. Its eigenvectors, later known as eigenfaces, are computed approximately by neural networks and will serve as the face description. Formally, given a set of n by m pixel training images represented as vectors of size m x n, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images are minimized. This set of optimal basis vectors is called eigenfaces since these are simply the eigenvectors of the covariance matrix formed from the vectorized face images in the training set. To detect the presence of a face in a scene, the distance between an image region and the face space is computed for all locations in the image. This distance is used as a measure of "faceness", resulting in a "face map". A face can then be detected from the local minima of the face map.

Article [15] (Sung and Poggio) is an example of distribution-based models that uses a multiplayer perceptron (MLP) network as a classifier applied to the distribution-based models for face/non-face patterns. The patterns are grouped into six face and six non-face clusters. Distances are computed between an input image pattern and the prototype clusters and will be used by MLP for classification. Principal component analysis and neural networks are used in numerous other literatures in this category.

A neural network is a powerful tool for solving pattern classification problems. It is not surprising to see that various neural network architectures have been proposed for face detection. The advantage of using a neural network is its capability to capture the complex class conditional density of face patterns. However, the network architecture has to be extensively tuned (number of layers, number of nodes, etc.) to achieve good performance ([2]).

Though the example methods mentioned above in each category are only selected samples, it is intended to provide sufficient background knowledge of the current status of the field. For a more complete review, readers may refer to [2]. After the publication of this survey, there have been several recent developments in this area, particularly the one by Viola and Jones ([15]) that is employed in our demo system. As claimed by the authors, this system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection. We will describe this particular method in detail, along with our system design, in the next chapter.

3. Demo System Design

3.1 Requirements of the demo system

This system shall have the capacity to detect and indicate the locations on images of human faces that are within a specified distance range of a web camera. After a specified number of seconds has elapsed, assuming that no face has been detected, the system will log off. Consequently, unauthorized users are denied access to the given computer.

The system should have a graphical user interface, in which a set of parameters related to the performance of the underlying computational algorithm will be displayed and can be manipulated by the user. A set of statistical information indicating approximately the current performance of the system is displayed in real-time. The user may adjust the allowed parameters and observe their effects on the system performance interactively.

This program is designed for machines running Microsoft Windows NT, 2000 and XP. It should be tested with desktop as well as laptop PC's, with different CPU speeds that are currently in use.

3.2 Main system components

The system consists of three main components: input, processing, and output units. An overview of these components is given below.

Input Unit: The system will acquire video images from a commonly used web camera. Logitech webcams were selected due to their popularity and availability. The current higher end model from Logitech for mass consumers is the Orbit model. This camera generates images of sizes at 640x480 or 320x240 pixels in either 24-bit or 32-bit color. The lower end model, Logitech Messanger, producing 320x240 color images of lower quality, is also supported by the demo system. The captured video frames will be subsequently displayed, via DirectX, to the user's screen in real time.

Processing Unit: The central task of this component is to detect faces appearing in each video frame. The detection method is based on the Viola and Jones' algorithm ([16]) that will be described in detail in the next section. This algorithm works with still images; thus, it does not use any dynamic information from a video sequence. This algorithm also works with grey-scale images – it does not make any assumption of skin colors. As a result, this algorithm is one of the most robust methods currently available.

The locations of objects suspected to be human faces will be recorded. Later, when a video image is finally rendered to the monitor, a red rectangle (or other effect) will enclose (or highlight) each suspected face.

No algorithm can be expected to perform flawlessly, especially in the case of face detection and recognition due to the challenging issues discussed in Chapter 2. Therefore, it is anticipated that this algorithm will mistakenly identify objects that are not actually faces, or will fail to detect objects that are faces. Though it is not practical to completely solve these two problems at the moment, an attempt to reduce them was made. The basic idea is as follows. The system examines more than one frame in order to determine if an object is likely a face or not. When an object presumed to be a face is discovered, the corresponding region in the previous frame is examined. If there were no faces found in this area before, then the current object is unlikely to be a face. As a result, the object is not marked as a potential face. Conversely, had there been faces present in this region in the previous frame, the likelihood that the current object represents a face is greater; in this situation, the system assumes that the object is a human face.

This entire process is summarized in Figure 1 in detail.

Output Unit: As mentioned above, items found in a scene that are determined be human faces are highlighted by a surrounding red rectangle (or accentuated by other means) in the resulting video output. Periodically, snap shot images from the video sequence are displayed to the screen. The user has the ability to modify the length of the interval between individual snap shots. Numerous statistics are also presented to the user. Such information reveals the inner workings of the algorithm and its performance, as well as reporting the estimated position of the user and countdown times.

3.3 The Viola and Jones' algorithm

Viola and Jones ([16]) developed a machine learning approach for visual object detection that is capable of processing images extremely rapidly and achieving high detection rates. This method has three distinguishing features.

The first is a learning algorithm, based on AdaBoost, which selects a small set of critical visual features from a larger set and yields extremely efficient classifiers. Features are usually a higher-level description of objects than pixel-level information. Features can be used to represent ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. In general, a feature-based system operates much faster than a pixel-based system. The three simple feature used in this system are reminiscent of Haar basis functions (Figure 2).

The first type has two rectangular features ((a) and (b) in Figure 2) that compute the differences between the sums of the pixels within two adjacent rectangular regions. They can be considered as the two first derivative operators, in horizontal and vertical directions respectively, at multiple scales. When the region size is one pixel, they are simple numerical approximations of the first derivative at each pixel. When the region size is greater than one pixel, we compute the gradient, or contrast, between two adjacent regions.

The second type is a three-rectangle feature ((c) in Figure 2) that computes the sum within two outside rectangles subtracted from the sum of the center rectangle. This feature can be considered as a second derivative operator at multiple scales.

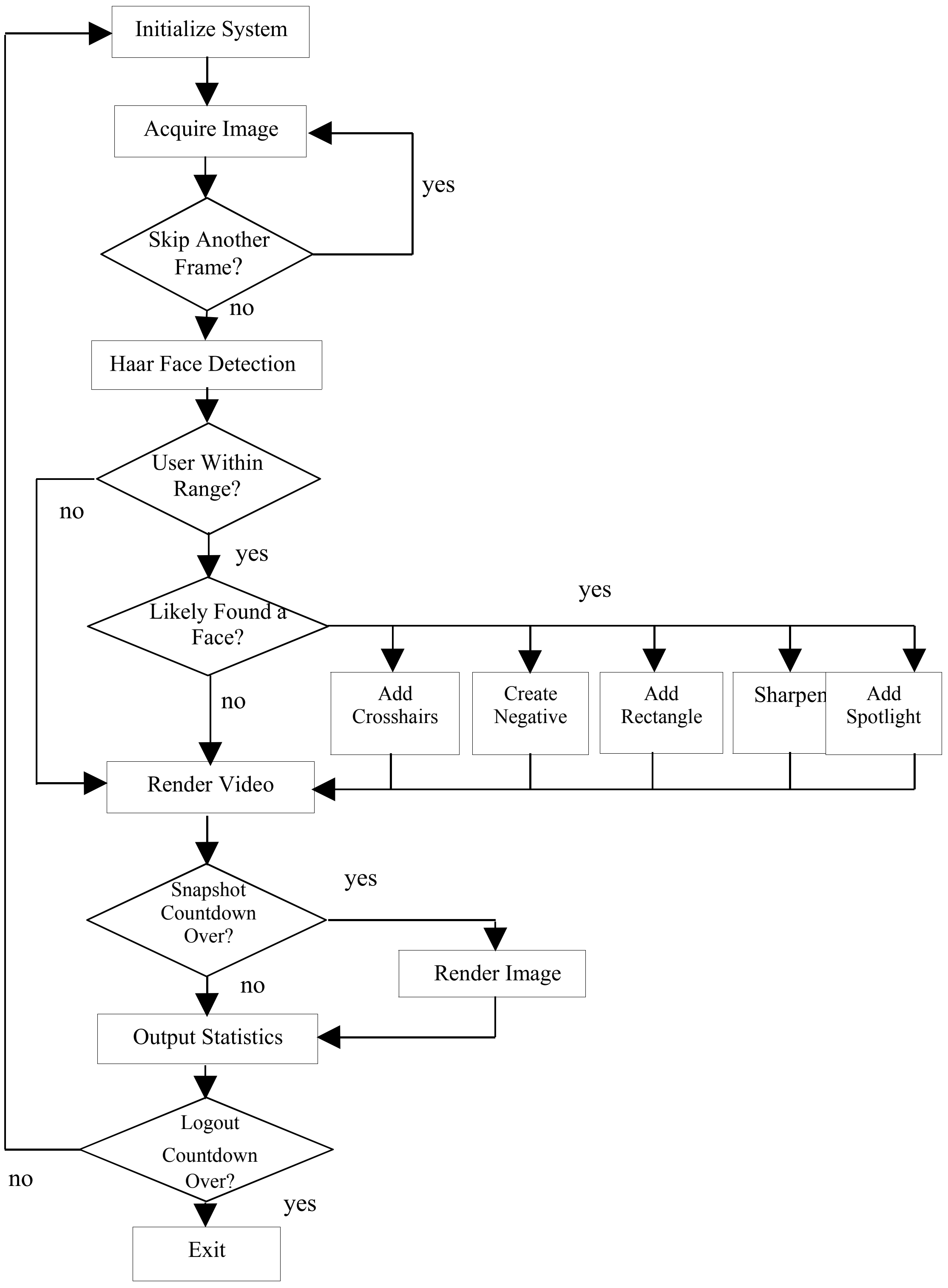

Initialize System
Acquire Image
Skip Another Frame?
yes
no
Haar Face Detection
User Within Range?
no
yes
Likely Found a Face?
yes
no
Add Crosshairs
Create Negative
Add Rectangle
Sharpen
Add Spotlight
Render Video
Snapshot Countdown Over?
yes
Render Image
no
Output Statistics
Logout Countdown Over?
no
yes
Exit

Figure 1. *System Diagram*

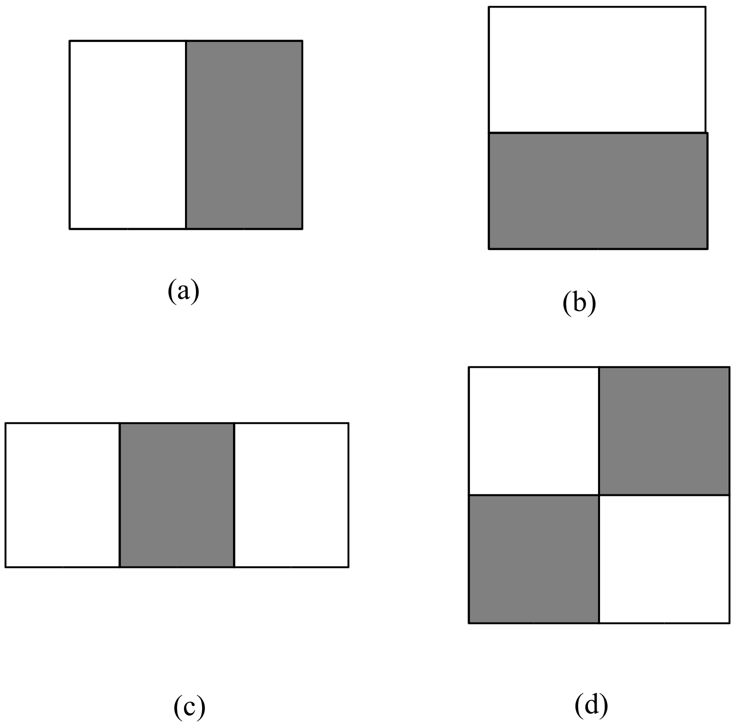


Figure 2. *Example Rectangular Features*

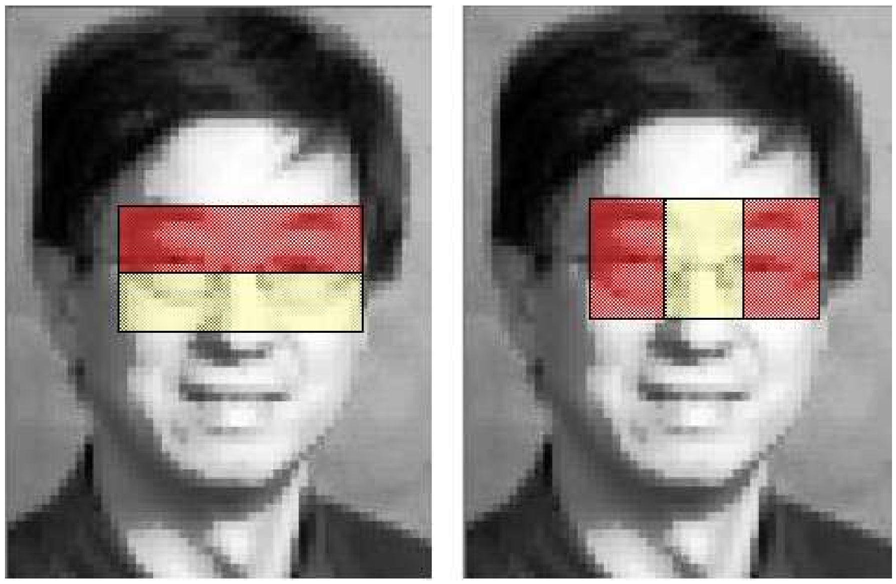


Figure 3. *Two Features Overlaid on a Typical Training Face.*

The third type is a four-rectangle feature ((d) in Figure 2) that computes the difference between diagonal pairs of rectangles. This feature can also be considered as a second derivative operator in a diagonal direction at multiple scales.

The set of rectangular features is very primitive, easy to compute, and has coarse responses to edges, bars, and other simple image structures. The intuitive meanings of these features is illustrated by the examples in Figure 3. The first and second features selected by AdaBoost are overlaid on a typical training face. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. This feature is based on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose. As reported in [16], a frontal face classifier constructed from 200 features yields a detection rate of 95% with a false positive rate of one in 14084. It should be noted that this algorithm is designed for detecting reasonably frontal faces because of the training images used for selecting features.

The above feature operators must scan through an image and the feature values are calculated at each position. Furthermore, because there is no assumption about the range of face sizes, these features have to be computed at all possible scales. For real-time applications, the computational efficiency of these multi-scale feature calculations will be the critical concern. This leads to the second distinguished feature of this system, the use of integral image.

The concept of an integral image was originally developed in computer graphics for fast multi-resolution sampling of texture images. This technique allows the features used by their detector to be computed very quickly and in constant time at arbitrary scales. The concept of integral image is based on the partial sum map (refer to Figure 4). Given an image $f(x,y)$ of size $M \times N$ ((a) of Figure 4), the corresponding integral image $f_{ii}(x,y)$ ((b) of Figure 4) is defined as:

$$f_{ii}(x,y) = \sum_{x' \le x, y' \le y} f(x',y')$$

That is, the value at the location (x,y) in the integral image, $f_{ii}(x,y)$, is the sum of the pixels above and to the left of (x,y) in $f(x,y)$. With an integral image, the sum of pixels of any rectangular region in $f(x,y)$, defined by the coordinates at the two corners (x_{min},y_{min}) and (x_{max},y_{max}) in (a) of Figure 5, can be computed in constant time through the information available in the corresponding integral image:

$$S(x_{min},y_{min},x_{max},y_{max}) = f_{ii}(x_{max},y_{max}) - f_{ii}(x_{max},y_{min}) - f_{ii}(x_{min},y_{max}) + f_{ii}(x_{min},y_{min})$$

Consider the four shaded rectangular regions (refer to (b) of Figure 5):

- $f_{ii}(x_{max},y_{max})$ is the sum of pixel values in all four rectangles in $f(x,y)$;
- $f_{ii}(x_{max},y_{min})$ is the sum of pixel values in the two upper rectangles in $f(x,y)$;

- $f_{ii}(x_{min},y_{max})$ is the sum of pixel values in the two left rectangles in $f(x,y)$; and
- $f_{ii}(x_{min},y_{min})$ is the sum of pixel values in the upper-left rectangle in $f(x,y)$.

The sum of pixel values in the lower-right rectangle, i.e. the region to be calculated, can be easily obtained by the total sum subtracted by the two upper and two left regions. Because the upper-left region is subtracted twice, it must be added back. Therefore, the sum of an arbitrary rectangular region in $f(x,y)$ can be calculated in constant time. This is the primary reason for the fast performance of this method.

The third distinguishing feature of this system is a method for combining increasingly more complex classifiers in a cascade that allows background regions of the image to be quickly discarded while spending more computation time on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism that, unlike previous approaches, provides statistical guarantees that discarded regions are unlikely to contain the object of interest. Though the new method is applicable to general object detection problems, the authors had a specific example in their mind - face detection. As reported in [16], a 38-layer cascaded classifier was trained to detect frontal upright faces. The number of features in the first five layers is 1, 10, 25, 25 and 50 respectively. Specifically, the initial one feature classifier was trained with a large collection of non-face examples, for the purpose of quick rejection of non-face regions in application. The remaining layers have increasingly more features. The total number of features in all layers is 6061.

As claimed by the authors, the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection. Lienhart and Maydt ([17]) extended the above work by introducing a set of rotated Haar-like features, and claim improved performance.

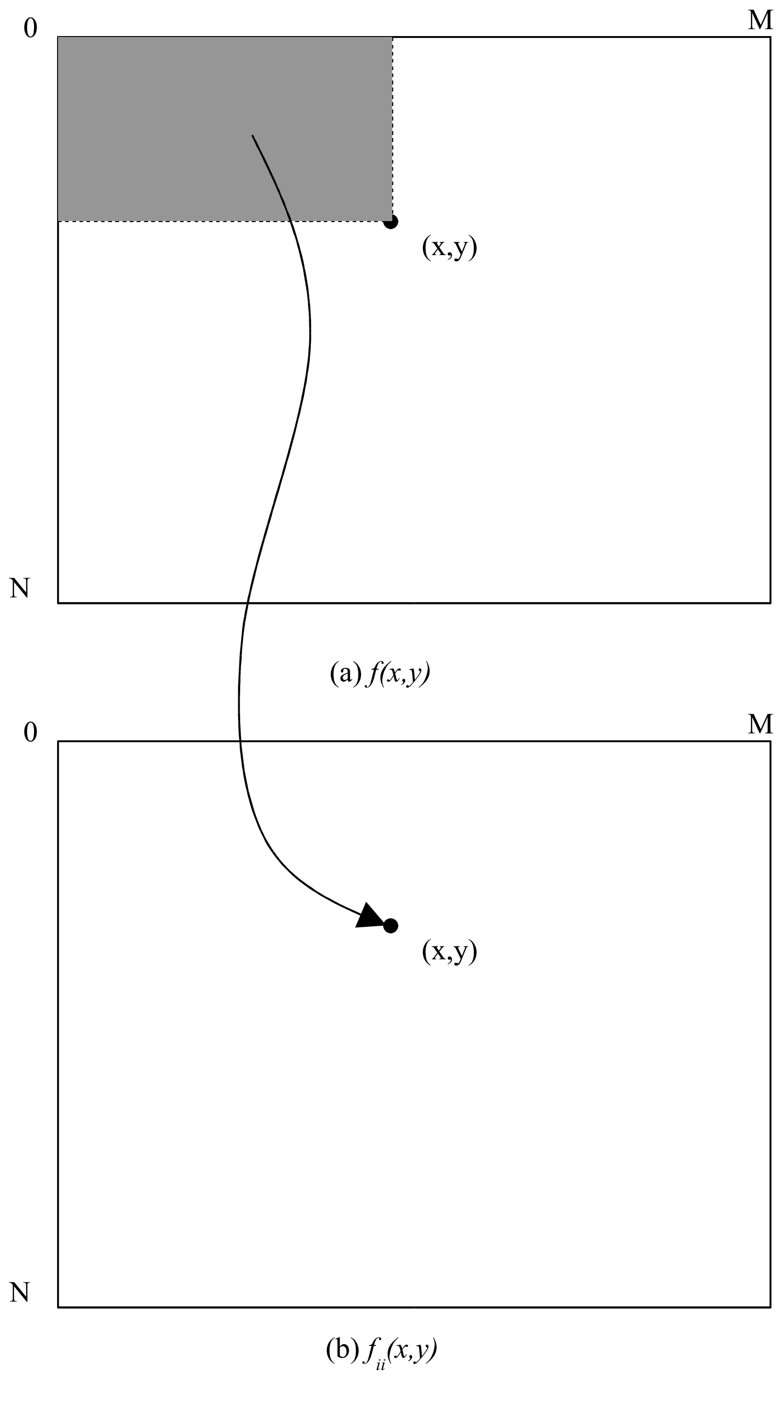
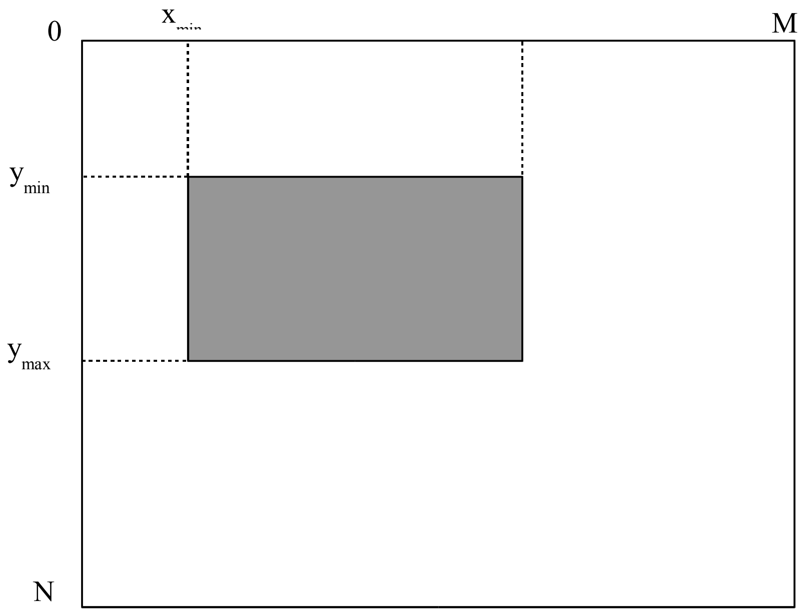
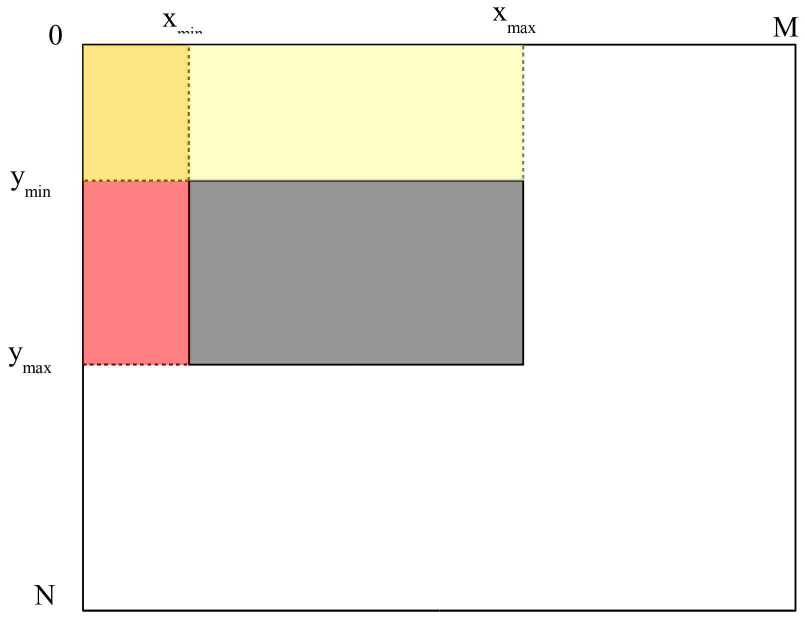


Figure 4. *The Concept of Integral Image.*



(a) *f(x,y)*

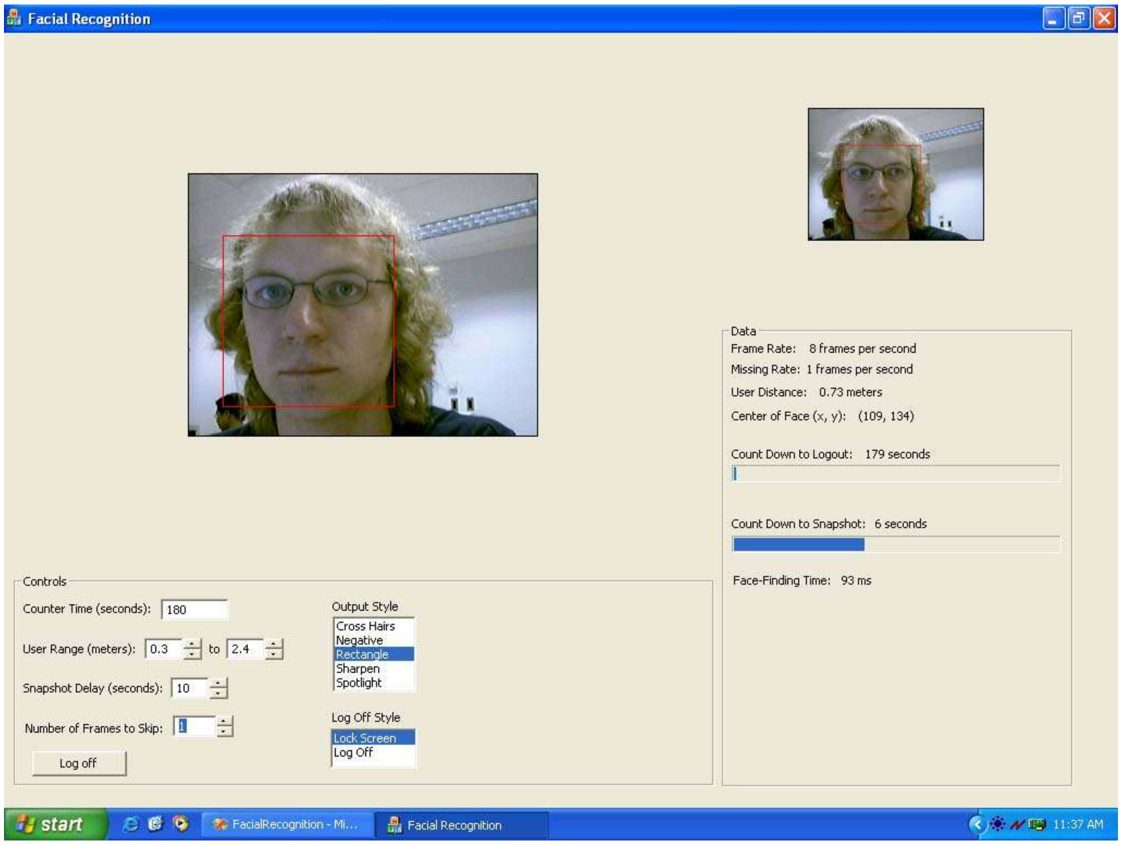


(b) $f_{ii}(x,y)$

Figure 5. *Calculation of Sum for Arbitrary Rectangular Region.*

3.4 System interface design

Figure 6 shows a snapshot of the system interface. The image on the top-left is the current frame captured from the video. The smaller image on the top-right is a still image, called a snap shot, refreshed periodically with a user adjustable interval. Other items in the interface can be divided into two groups: controls and statistical data.



Data
Frame Rate: 8 frames per second
Missing Rate: 1 frames per second
User Distance: 0.73 meters
Center of Face (x, y): (109, 134)
Count Down to Logout: 179 seconds
Count Down to Snapshot: 6 seconds
Face-Finding Time: 93 ms

Controls
Counter Time (seconds): 180
User Range (meters): 0.3 to 2.4
Snapshot Delay (seconds): 10
Number of Frames to Skip: 1
Log off

Output Style
Cross Hairs
Negative
Rectangle
Sharpen
Spotlight

Log Off Style
Lock Screen
Log Off

Figure 6. *A Snapshot of the System Interface*

Controls

1. Counter Time – This control allows the user to input the number of seconds that must pass before the system exits and logs out. For example, a value of 15 would mean that the system would log out in 15 seconds, if during the next 15 seconds no face is found.

The counter starts counting down when no face is detected in the current video frame. If a face is detected at any time during this countdown, the countdown will reset to the Counter Time value and the countdown process begins again.

2. User Range – These two input fields are the minimum and maximum distances that a user may be from the camera, in meters, in order for the system to detect that individual. A face that is found at an estimated distance less than the minimum will not be reported. Likewise, one that is farther than the maximum distance will be ignored.

The values in these two fields can be modified in two ways. First, the user can directly enter a numerical value in either field. An alternative way is to use the up/down buttons to the right of each field to interactively increase and decrease these values. Values of 1.5 and 3.0, for instance, mean that a user would have to be at least one and half, but no more than three, meters from the camera in order to be detected by the camera.

3. Snap Shot – The value given in this field is the number of seconds that are allowed to pass between each snapshot that is taken. By adjusting it, one can create still images more or less often. For instance, a value of 5 would mean that the still image (i.e. snap shot) from the camera is refreshed every five seconds.
4. Logoff – This button will allow the user to manually log out of the operating system. After pressing it, the user is prompted to confirm that they indeed wish to exit. If so, the system immediately logs out. Otherwise, there is no effect.
5. Logoff Style – There are two optional actions upon logoff. One is to logoff the user from the Windows operating system. In this case, all user applications currently running are terminated. The other is to lock up the screen, similar to the automatic screen saver function of Windows. In this instance, all user applications currently running are not terminated.
6. Output Style – This control allows the user to select the highlight style of detected faces in the images seen in the video output rectangle.

- The default selection, *Rectangle*, causes the system to draw a red bounding rectangle around any face detected. The remaining styles produce various effects.
- Should one select *Cross Hairs*, then a red circle and black cross hairs would be drawn over any faces.
- The *Negative* option draws all non-facial regions in negative colors (much like the negative of a black and white image).
- By selecting *Sharpen* a user can request that the system blur all non-facial regions; thus, this effectively “sharpens” the circular regions around any faces present in the scene.
- Finally, the *Spotlight* option causes all non-facial regions to be darkened. This gives the impression that lights are being applied to the faces in the scene.

7. Number of Frames to Skip – The final field is the *Number of Frames to Skip* control. In order to reduce the overhead and amount of processing time needed by this system, the user has the option not to search for faces in each frame captured by the camera. Instead, the system may skip over a specified number of frames. The result is a significant improvement in the speed and performance of the program. This also means that there is more processing time and memory available to other applications that are also running. As an example, if one were

to enter a value of 5, then the system would search for faces in only every fifth image taken by the camera.

Statistical data

1. Frame Rate – This value indicates the number of individual frames that are being processed by the system per second. For instance, a value of 8 would mean that the program is able to receive, examine, search and output eight frames in a single second. This value depends on the performance of the PC system. There are a number of factors that can affect this value. The primary determining factor is the CPU speed. Higher frame rates can be expected on faster machines. There are also many other factors, such as memory size and the number of other applications running at the same time. The displayed frame rate gives the effective performance of the system in real-time.

2. Missing Rate – This value indicates the number of frames per second in which no face is detected. It is the primary information related to the detection rate of the system. This value must be interpreted carefully. If at least one face actually appears in the video, then this value is the true missing rate. However, if no face appears in the video, then this value would not have any meaning. That is, even though this value is quite large in the later case, it should not be interpreted as the "missing" rate.

3. User Distance – This field provides an estimate of the user's distance, in meters, from the camera. A value of 2.75 would mean that the user is estimated to be approximately 2.75 meters from the camera.

   This value is estimated based on the size of the detected face in the image. An assumption is made about the average physical size of an adult face. The actual physical size of an individual face, of course, varies from one person to another. Therefore, a smaller physical face will cause a larger estimated distance value, and vice versa. This value is also affected by the calculated face size on the image that is particularly inaccurate if the face is very close to or very far from the camera. Our experiment shows that this value would be less accurate when a user is very close (half a meter or less) or extremely far away (over ten meters) from the camera.

4. Center of Face – Two values are given in this field. They represent the *xy*-coordinates of the center of the largest face (the user). This position is relative to the upper-left corner of the video output rectangle that is given as point (0, 0). Consider as an example, the point, (50, 75). According to the system, the center of the user's face is about 50 pixels to the left of the upper-left corner of the video screen, and 75 pixels down from the upper-left corner.

5. Count Down to Logout – This is the number of seconds remaining before the system shuts down, assuming that no face is found before the counter reaches zero. For instance, a value of 60 would mean that the system would log out in 60 seconds, unless a face is detected in the next minute. The progress bar below gives a visual indication of the amount of time remaining before logoff.

6. Count Down to Snapshot – Similar to the previous item, this field displays the number of seconds remaining before the next snapshot is taken. Once the counter reaches zero, a snapshot is taken by the camera and is displayed in the still image rectangle. A progress bar below this counter shows, visually, the amount of time remaining before the next snapshot is taken.
7. Face-Finding Time – Another important measure, this field estimates the amount of time, in milliseconds, that was used by the face-finding procedure to locate faces in the current frame. Assume that a time of 30 ms is given. This means that the face-finding routine needed 30 milliseconds of CPU processing time to locate all of the faces in the current frame. The purpose of this information is to demonstrate the real-time speed of the software.

4. Implementation and Technical Issues

4.1 Overview of OpenCV library

The first issue is that of capturing video frames from a webcam. This task would be easy if a software development Kit (SDK) from the manufacture of the webcam were available. Unfortunately, those SDKs are usually available to only commercial product developers and the license fee can be very expensive. Furthermore, we may want the tool to support more than one model of webcams. This requires a generic interface to video devices. The solution to this problem in the demo system is a function provided in OpenCV, Intel's Open Source Computer Vision Library [18]. In addition, the state-of-the-art face detection methods ([15, 16]) are implemented in OpenCV and are available for use.

OpenCV library is an open source software package aimed at real-time computer vision tasks. This library is intended for use, incorporation and modification by researchers, commercial software developers, government and camera vendors as reflected in the license [17]. The beta version 3.1of OpenCV is used in this system.

The OpenCV library is divided into a number of different groups:

Image Processing and Analysis: this includes functions for edge detection, color conversion, and histogram generation and analysis.

Structural Analysis: this deals with two-dimensional polygons, edges, and points.

Motion Analysis and Object Tracking: this has optical flow algorithms, as well as general object-detection functions.

Object Recognition: this deals with Eigen objects, and hidden Markov Models, two different ways of recognizing objects.

Camera Calibration and 3D Reconstruction: this includes methods for detecting and compensating for camera distortions, and putting two-dimensional video into three-dimensional space.

4.2 Description of OpenCV functions used in the demo

In this demo system, we have used a number of functions from this library. The functions used can be grouped into two categories - video frame capturing and face detection.

Video frame capturing

CvCaptureFromCam(): This function specifies which camera to use as the capture device. The input parameter, an integer, is the ID of the desired camera. It returns a pointer to a CvCapture structure.

CvGrabFrame(): This routine secures a frame from the capture device specified by the input parameter. The grabbed frame is stored internally.

CvRetrieveFrame(): This function is used to return the secured frame from the capture device specified by the input parameter. It returns a pointer to an IplImage structure.

The IplImage structure, from the OpenCV library, includes the raw image data, and the attributes associated with this data (such as height, width, number of color channels, and bits per pixel). The structure also has some other things that we did not make use of, such as Rectangle of Interest, or Channel of Interest.

CvMirror() (cvFlip() in the OpenCV documentation): It is used to flip the image vertically, so that it is right-side up. This function takes three parameters - a pointer to the source image, a pointer to the destination image (done in place if NULL), and which direction to flip it.

Using byte pointers, the image data is then copied from the IplImage structure into our own byte array, so that the image exists in two formats for ease of processing.

Face detection

CvLoadHaarClassifierCascade() loads a trained cascade classifier from a file to a pointer. The default input parameters are used, and the function returns a pointer to a CvHaarClassifierCascade structure.

CvCreateHidHaarClassifierCascade() makes the cascade classifier that was just loaded from file usable and ensures that it is optimized.

CvReleaseHaarClassifierCascade() deallocates memory and clears the pointer to the file cascade.

CvClearMemStorage() clears the memory that the input parameter is using. The parameter is a CvMemStorage structure type. This is simply a general memory storage structure from OpenCV.

CvHaarDetectObjects() detects objects using the cascade classifier that is given to it. This function takes the image, the cascade classifier, the memory storage, and three other parameters that can be used to speed-up the search, or eliminate some unwanted results. A pointer to a CvSeq is returned.

CvGetSeqElem() gets an element from the OpenCV array type CvSeq. In this case, the elements are cast to the type CvRect.

CvRect is a structure with four attributes - x, y, height, and width. Throughout the program, this structure is used to represent faces.

4.3 Technical problems encountered during the experimentation and solutions

Four major technical problems were encountered during the development of this project.

False detection caused by artifacts

The largest and most significant problem was dealing with the output that came from the cvHaarDetectObjects function. This output was generally very good, but there were always a number of artifacts detected. That is, it detected faces where there were no faces actually present. Generally though these false faces would only last one frame, and they were usually the product of the camera adjusting to the light or moving around. A simple solution that we developed to deal with these artifacts was to keep a list of the faces detected in the last frame, and to compare it to the list of those detected in the current frame. In this way, if a face was detected, and there was no face detected anywhere near it in the last frame, it could be discounted. This approach successfully eliminated most artifacts in the raw results from cvHaarDetectObjects function. This, however, introduced a limit on the speed a person could move while being detected. Overall, this solution has ensured that the logout counter worked properly.

Multiple video capturing devices

Another problem was encountered when there were multiple cameras or capture devices connected to the computer. This problem came from the cvCaptureFromCam function. When this function is called, an integer parameter is passed to it specifying the ID of the camera; if it is –1, OpenCV chooses the default camera. OpenCV, however, does not provide functions for finding the IDs of the cameras attached to the computer.

A problem occurred when we installed more than one different webcam driver on a Windows system. When the camera connected to system is replaced by the other, the demo system may have problems capturing video frames from the new camera, as the previous working driver was set as the default and is still selected. A similar problem also arose in the case in which another video capture card, such as one for video capturing/editing applications, was installed in the operating system.

The most appropriate solution was to call the function twice with the parameter 0. This popped out a dialog box that let the user choose which camera to use. Since this clutters up the dialog, and because it is rarely used, this was made into a separate utility. If there are problems with the camera selection, one should exit from the facial recognition program, run the provided "choose-camera" utility, select the correct camera, and enter the facial recognition program again.

Dynamic video frame size

Also related to the webcam settings, there was a problem with the size of the frames obtained from the camera. Low-end webcams usually have a fixed frame size. However, many middle- and high-end products offer variable frame sizes, typically 640x480 and 320x240. More intelligent products may even detect the speed of the connection (such as USB 1 or USB 2) and automatically select the frame size. But, control programs always have the options to select a desired frame size. OpenCV does provide a function for changing these settings for the camera, but it does not work properly with our Logitech Orbit camera. A clever solution to this problem would be to include interface functions to the drivers for lower level controls. Unfortunately, the required software is typically available in an SDK from the manufacture and it was not available to us during the development for the reason explained earlier.

The problem then is that the size of the frames obtained from the camera could not be guaranteed to be of a fixed size. A simple solution is to have a maximum resolution that is allowed, 640 X 480, and if the camera returns images that are smaller than that, it is centered in the video frame region and the rest of the space is filled with the background color of the window.

Pixel color formats with DirectX

One problem was encountered while using DirectX to draw the output. While writing the individual pixels to the screen, Windows' different bits/pixel settings need to be taken into account. Currently only 24-bit and 32-bit color formats are supported. If other lower-color qualities are used, then the face detection and logout will work properly, but the video output will be garbage. Therefore, the user needs to set the screen display mode to either 24-bit or 32-bit before running this application.

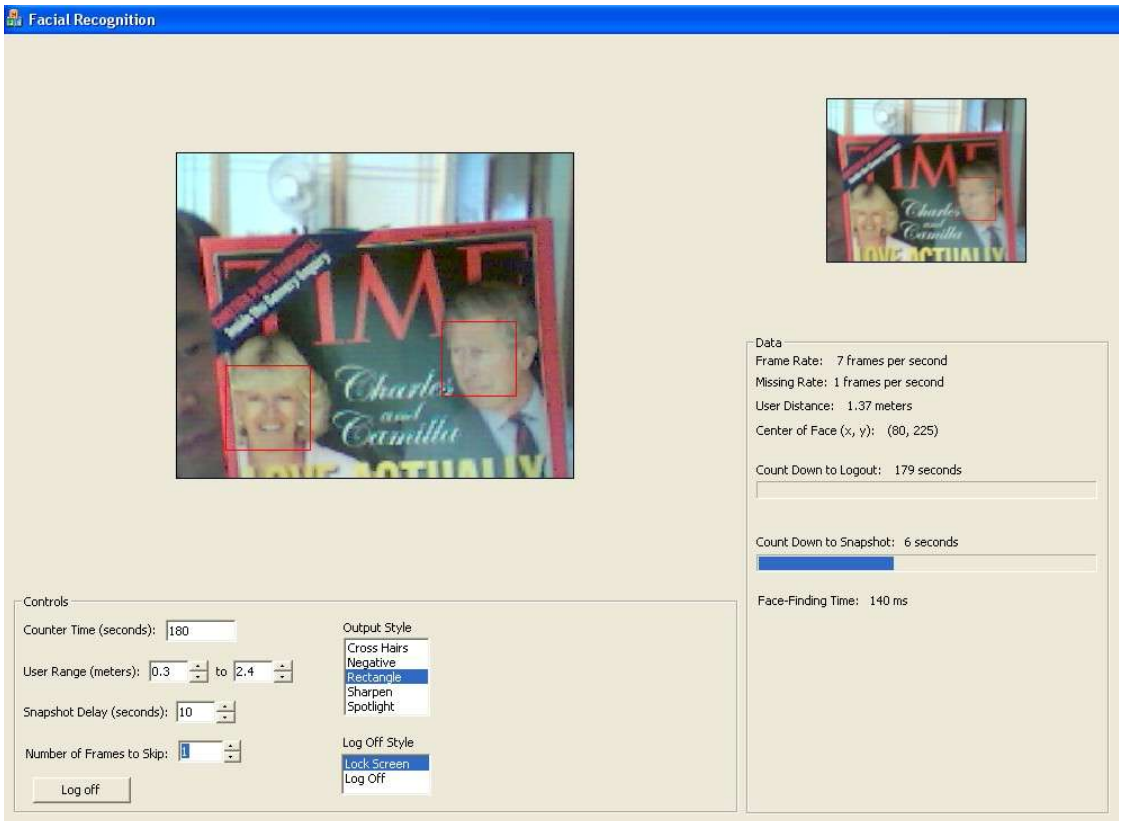
4.4 Performance discussion

The video frame processing speed varies depending on the CPU speed of the PC being used. In this system, we search for faces at any scale and any location in each frame. The information from the previous frame is not utilized in searching the subsequent frames. The frame processing rate was as high as 15 frames per second on a dual-processor PC with CPU speed at 2 GHz, and as low as 5 frames per second when a 1.6 GHz single CPU PC was used.

There is great potential to improve the frame processing rate in the future. First, the knowledge from the previously processed frames can reduce the search areas for the subsequent frames. Given that a face has been detected in the previous frame with a known size and location, the search in subsequent frames can be restricted to the neighboring area using similar feature sizes. In addition, according to the user range set in the control panel, the feature size can be narrowed down to a corresponding range. Although these improvements are highly achievable, they, however, require modification and customization of the detection algorithm. Therefore, they were not implemented in this project due to limited time and budget concerns.

The detection accuracy is related to the missing rate. We have observed an average error rate (defined as missing rate/frame rate) of about 30%. That means, considering a single frame, the chance to miss face(s) in the frame is around 0.3. It seems quite poor. But, if we consider two consecutive frames, the chance to miss face(s) consecutively is around 0.3 x 0.3 = 0.09. If three consecutive frames are considered, the chance to miss face(s) in a row will become 0.027. Therefore, the overall accuracy is sufficient for this type of application.

Due to the nature of the Haar function used for feature extraction in this system, there are obvious limitations to this method. For example, printed faces (such as a magazine cover shown in Figure 7) are also reported as being faces in the scene. The current method does not have the capability to distinguish between real 3D faces and printed 2D faces. Additional information and computation are required to make such a classification. Furthermore, some simple sketches (Figure 8) can fool the system.



Data

Frame Rate: 7 frames per second
Missing Rate: 1 frames per second
User Distance: 1.37 meters
Center of Face (x, y): (80, 225)

Count Down to Logout: 179 seconds

Count Down to Snapshot: 6 seconds

Face-Finding Time: 140 ms

Controls

Counter Time (seconds): 180
User Range (meters): 0.3 to 2.4
Snapshot Delay (seconds): 10
Number of Frames to Skip: 1

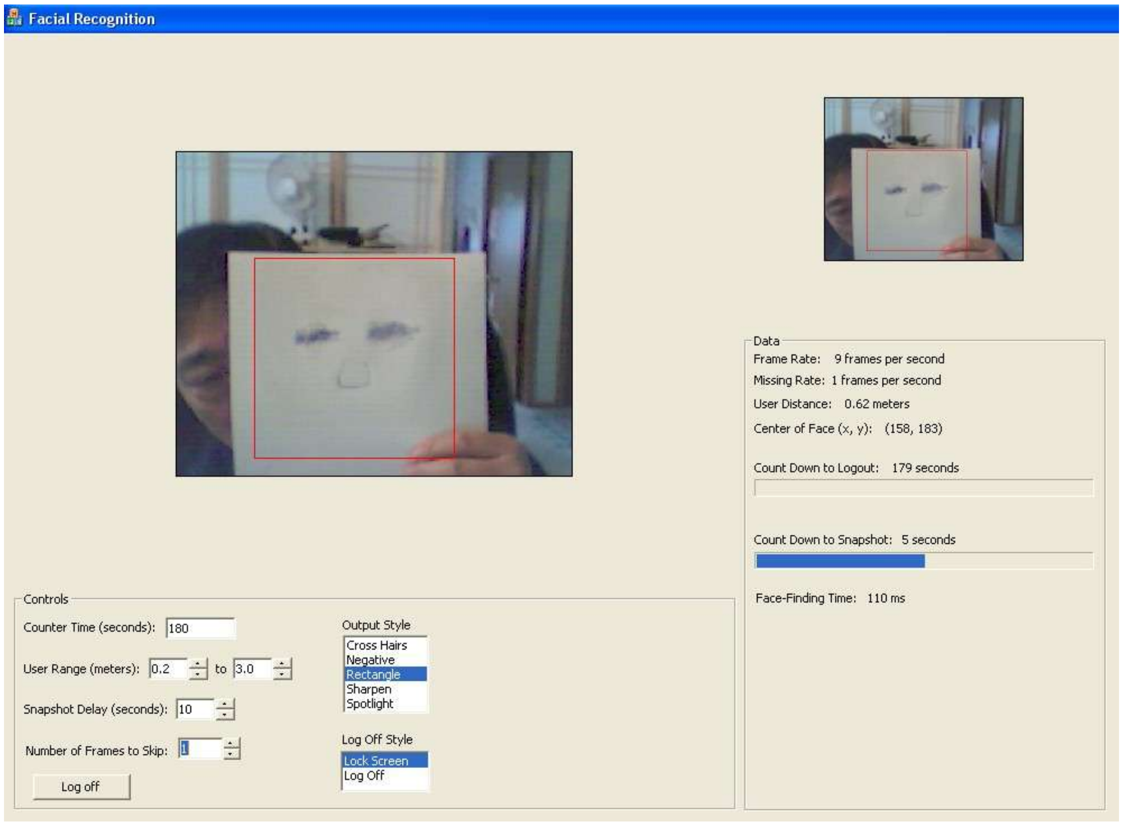
Output Style: Cross Hairs / Negative / [✓ Rectangle] / Sharpen / Spotlight

Log Off Style: [✓ Lock Screen] / Log Off

Figure 7. *Test with 2D Printed Faces*

The employed detection algorithm also performs poorly on dark-skin faces because the intensity contrast between different facial regions, that are the basis for the Haar functions, is not significant.

For the same reason above, illumination will affect the performance of the detection algorithm. If the illumination condition is poor, the faces will appear very dark in the input image. Consequently, features being detected by the Haar functions could be very weak, leading to a higher missing rate.



Counter Time (seconds): 180

User Range (meters): 0.2 to 3.0

Snapshot Delay (seconds): 10

Number of Frames to Skip: 1

Output Style: Rectangle

Log Off Style: Lock Screen

Frame Rate: 9 frames per second

Missing Rate: 1 frames per second

User Distance: 0.62 meters

Center of Face (x, y): (158, 183)

Count Down to Logout: 179 seconds

Count Down to Snapshot: 5 seconds

Face-Finding Time: 110 ms

Figure 8. *Test with Simple Sketch*

5. Summary

- The proposed demo system has been developed by the authors within the specified deadline.
- The demo system meets all the predefined requirements and even exceeds some of them.
- Performance of the underlying face detection algorithm can be interactively observed by users by adjusting control parameters.
- This system clearly demonstrated the feasibility of real-time continuous user monitoring of information systems requiring high-level security.

Logical steps, following this project, will be, but are not limited to, the following items:

- To develop an algorithm that can effectively distinguish between a real 3D face and a 2D printed face through dynamic analysis of video frame sequence, or by using two webcams placed at different view points combined with stereo vision principles.
- To improve the frame rate by developing a local dynamic tracking algorithm that is used in subsequent frames once one or more faces have been reliably detected in a frame.
- To include face recognition into the system, such that it not only detects faces, but also recognizes authorized faces. Based on the critical analysis given in [1], a robust method is likely to employ 3D models and techniques alone, or together with 2D image analysis techniques.

Finally, we would like to emphasize that the delivered demo system and the proposed functionalities for future development is an enabling technology that is not only useful for the particular application addressed in this project, but also for many applications that are concerned with continuous monitoring.

References

1. W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face Recognition: A Literature Survey." *ACM Computing Surveys*, Vol. 35, No. 4, December 2003, pp. 399–458.
2. M. Yang, D. Kriegman, and N. Ahuja. "Detecting Faces in Images: A Survey." *IEEE Trans. PAMI*, Vol. 24, No. 1, January 2002, pp.34-58.
3. H. Yang and T.S. Huang. "Human Face Detection in Complex Background." *Pattern Recognition*, Vol. 27, No. 1, 1994, pp.53-63.
4. C. Kotropoulos and I. Pitas. "Rule-Based Face Detection in Frontal View." *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Vol. 4, 1997, pp.2537-2540.
5. S. Sirohey. "Human Face Segmentation and Identification." *Tech. Report CS-TR-3176*, Univ. of Maryland, 1993.
6. H. Graf, T. Chen, E. Petajan, and E. Cosatto. "Locating Faces and Facial Parts." *Proc. 1st Int. Workshop Automatic Face and Gesture Recognition*, 1995, pp.41-46.
7. D. Cai and K.N. Ngan. "Locating Facial Region of a Head-and-shoulders Color Image." Proc. 3rd Int'l Conf. Automatic Face and Gesture Recognition, 1998, pp.124-129.
8. T. Sakai, M. Nagao, and S. Fujibayashi. "Line Extraction and Pattern Detection in a Photograph." *Pattern Recognition*, Vol. 1 1969, pp.233-248.
9. V. Govindaraju, D. Sher, R. Srihari, and S. Srihari. "Locating Human Faces in Newspaper Photographs." *Proc. IEEE Conf. CVPR*, 1989, pp.549-554.
10. A. Tsukamoto, C. Lee, and S. Tsuji. "Detection and Tracking of Human Face with Synthesized Templates." *Proc. 1st Asia Conf. Computer Vision*, 1993, pp.193-186.
11. A. Samal and P. Iyengar. "Human Face Detection using Silhouetes." *Int. J. Pattern Recognition and Artificial Intelligence*, Vol. 9, No. 6, 1995, pp.845-867.
12. J. Miao, B. Yin, K. Wang, l. Shen, and X. Chen. "A Hierarchical Multiscale and Multiangle System for Human Face Detection in a Complex Background Using Gravity-Center Template." *Pattern Recognition*, Vol. 32, No. 7, 1999, pp.1237-1248.
13. A. Yuille, P. Hallinan, and D. Cohen. "Feature Extraction from Faces Using Deformable Templates." *Int. J. Computer Vision*, Vol. 8, No. 2, 1992, pp.99-111.
14. T. Kohonen. *Self-Organization and Associative Memory*. Springer, 1989.
15. K. Sung and T. Poggio. "Example-Based Learning for View-based Human Face Detection." *IEEE Trans. PAMI*, Vol. 22, No. 1, 1998, pp.39-51.
16. P. Viola and M. Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features." *TR-2004-043*, Mitsubishi Electronic Research Laboratory, May 2004.
17. R. Lienhart and J. Maydt. "An Extended Set of Haar-like Features for Rapid Object Detection." *Proc. Int. Conf. Image Processing*, 2002, pp.900-903.
18. http://www.intel.com/reserch/mrl/research/opencv/, as access on March 31, 2005.

List of symbols/abbreviations/acronyms/initialisms

DND	Department of National Defence
CFI	Canada Foundation for Innovation
NMSL	New Media Studio Lab
PC	Personal Computer
DirectX	An advanced suite of multimedia application programming interfaces by Microsoft
OpenCV	Intel’s Open Source Computer Vision library
SDK	Software Development Kit
ID	Identification
3D	Three-dimensional
2D	Two-dimensional

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.) Department of Computer Science University of Regina Regina, Saskatchewan S4S 0A2	2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable) UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Automatically Log Off Upon Disappearance of Facial Image Scientific Report (U)

4. AUTHORS (Last name, first name, middle initial)

Yang, Xue Dong ; Kort, Peter ; Dosselmann, Richard

5. DATE OF PUBLICATION (month and year of publication of document) March 2005	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.) 37	6b. NO. OF REFS (total cited in document) 18

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DEFENCE R&D CANADA - OTTAWA
3701 Carling Avenue, Ottawa, Ontario, K1A0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant) 15BF27	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written) W7714-4-0-9131
10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC Ottawa CR 2005-051	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
(   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
(   ) Distribution limited to government departments and agencies; further distribution only as approved
(   ) Distribution limited to defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)
13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual). The ability to prevent unauthorized users from gaining access to classified or sensitive data is crucial to many organizations, especially the military. There has been significantly increased attention in recent years in biometrics technology in order to provide more secured access controls to information systems. Different technologies and various products have been developed to authenticate users at entrances. However, once a user has logged on to a system, there is currently no mechanism to monitor the user's identity. It is important to ensure that the user, throughout the communication, is the authorized user for high-security applications. The objective of this project is to develop a demonstration system that will automatically log off a PC when the user's face disappears for an adjustable time interval. In this report, a brief overview of face detection technologies is provided. The particular neural network-based face detection algorithm employed in the demo system is discussed in more detail. It is followed by the design of the demo system, and discussion of technical issues encountered during the development.
14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.) Face Detection, Face Tracking, Auto Log off, Neural Network, Video-based Image Processing